Ceng492 Graduation Project

The Bride Project

The Bride Software Configuration Management Plan

Presented by Meda

Ankara, 2005

I. Table of Contents

1. Introduction

During the time of the software development we will be making changes to our original plans. Software Configuration Management Plan is developed so that we can identify the change, control the change, make sure the plan is implemented correctly and to make sure that we report the change to others.

1.1 Purpose

The purpose of this document is to establish the methods to be used to achieve the objectives of the Software Configuration Management (SCM) process, for “The Bride” program with the following items:

- Development of the project according to requirements, design, the schedule and coding standards presented.
- Testing & Debugging the project according to “The Bride - Test Specifications”.

In this plan, the implementation of software configuration management process, process activities, tasks, responsibilities, process inputs and outputs will be explained. Also, this plan includes the schedule of configuration management activities and the responsible team members for conducting software configuration activities and relationships among other team members. The Project Team, team leader and SCM, shall use this plan to achieve the configuration management objectives detailed in this plan.

1.2 Scope and Intent of SCM Activities

The main purpose of SCM is to make, report and track any changes made to the original software development plan. It is applied throughout the software development process and will help us to keep track of changes and also help us go through and make changes. SCM procedures will give us a good map out of the software so that if we are need to make more changes it will be relatively easy to do so. SCM will maximize productivity by minimizing mistakes. For SCM to be successful, all the members of software production team will have to take time to report the changes that they think are necessary and/or to notify others of changes that they may have made. This is sort of boring and time-consuming work, but it is very important.

SCM activities are developed to

- Identify change
- Control change
- Ensure that change is being properly implemented
- Also have a way to document the change.

1.3 SCM organizational role

Since we have rather small software development team, each member of the team will accept responsibility for software configuration management. This is necessary since there are only five members in the team. If one of the member reports changes remaining four members have to take up a job of authorizing change and to ensure that change is properly implemented. This will reduce or eliminate confusion between the team members regarding changes with the software.
When a need for change within the documents identified as software life cycle data that are baselined occurs, the team decides on the solution and, the team leader assigns the tasks that have to be realized to the team members. When the document is updated, it is controlled & approved by the team leader, and the new version of the related document is released by the configuration manager. The software life cycle data and the related documents are provided in Appendix part of this plan.

2. SCM Tasks

2.1 Life Cycle Data Control

The Bride plans shall identify the tools and data to be used to produce the software and data items. Unless otherwise specified in this document, Configuration Management activities shall be the same across all Bride units.

Software life cycle data will be subject to the following activities:

- Configuration Identification
- Baselines and Traceability
- Problem Reporting
- Change Control Tracking, Identification & Integrity
- Change Review
- Configuration Status Accounting
- Archive, Retrieval, and Release
- Protection against unauthorized changes
- Media Selection, Refreshing & Duplication
- Software Load Control
-

The following software life cycle data shall be handled:

- System and Software Requirements Data
- Software Design Description
- Software Configuration Index
- Source Code
- Executable Object Code
- Software Development Plan
- Software Configuration Management Plan
- Software Quality Assurance Plan

- Software Test Specifications and Procedures
- Software Coding Standards
- Software Configuration Management Records
- Software Quality Assurance Records
- Verification Results

In this section we will try to detail all-important SCM tasks and will assign responsibilities for each. All of the SCM tasks will be performed by three members of the software development team members. All the changes that affect the use of the software will be discussed with entire team.

2.2 Responsibilities

The Team Leader is responsible for scheduling and coordinating the activities of the entire program and preparing progress reports for management reviews. He ensures that the changes from external systems are communicated to the related members and also change control activities among the software engineers are performed.

The Software Development Team is responsible for generating the software requirements, design, code, and deployment of the program. The Software Development Team is also responsible for verification of the Verification Process results.

The System Integration & Test Team is responsible for reviewing code and documents that require independent verification, generating and executing tests, verifying the results of the testing, and confirming proper test coverage.

Configuration Manager is responsible for the control, release and storage of project documents and work products, as well as configuration management of the released documents. Configuration Manager is also responsible for the assignment, management and resolution of any Software Change Requests (SCR), control and management the baseline definition process, keeping all software products defined in the Configuration Item Identification Index under configuration control, preparing the monthly configuration status reports, saving the baselines and revisions of software products under the configuration control tool (“baseline” refers to a particular “snapshot” in time during development, i.e. a particular build, an item can be under CM control, and not yet be baselined.

The Software Quality Assurance Representatives (SQAR) are responsible for independent quality assurance reviews and audits to assure that the identified technical and procedural plans and standards are followed throughout the software life cycle which will be different individuals from the developer of each module providing independence.

2.3 Identification

This Configuration Management Plan (CMP) establishes the overall plan for the Configuration Management requirement for the Bride Software, the components of the

system, Computer Software Configuration Items (CSCIs) used during total system life cycle of the project and supports the Configuration Control Procedures.

2.1.1 Activities

1. Identify change

If during the software development phase a team member suggests a change in the software then we need to have the team work on the suggestion and to figure out if the change is necessary and is justified.

2. Approve change

We want to be able to have control over any change within the software. We can not afford to have one member of software development team think of a change or and implement it without telling any other member of the team. This can create huge technical problems for the software. We want to develop rules so that no member of the team will think of and implement change without permission of other members. We will be using the change request report form to suggest changes in the software. Below is the link to the page that contains Change Request Report generator and the picture of it.

3. Ensure that change is being properly implemented

We want to have team members looking over the change. Since all the teammates will be working separately, it is possible to have made mistake in implementing the change. To make sure this doesn't happen, we want to set up times when team members will look over the change that other members have implemented and to finalize the change.

4. Document the change.

Since change has to be documented from the time that a team member suggests change to the time change is finalized, we will end up with extensive documents. We will be using the change request form developed by Delta group to request change. To approve a change we will be using change report form, which will be submitted to change control panel (remaining member of the software development team).

5. Baselines and Traceability.

The objective of this activity is to establish baselines for CIs for further software life cycle process activity and to allow reference to, control of, and traceability between CIs.

Baselines shall be identified using appropriate version labels, which are associated with the CIs under consideration. Once a baseline is established, change control activities shall be followed to develop a derivative baseline. The baseline information for CIs shall be provided in the Software Configuration Index, Bride. CIs will be baselined at the times, prior to entry into the next life-cycle phase that is when the Project has reached the state described in the plans as the following build (baseline).

6. **Archieve, Retrieval and Release.**

Old the versions of the documents identified as software lifecycle data will be archived using CVS, by the control of Configuration Manager.

7. **Protection against unauthorized changes.**

The team members will only be able to make possible changes assigned to them. Possible changes to other system files will not be allowed through the use of CVS.

8. **Media selection, refreshing and duplication.**

9. **Software Load Control.**

Activities related to the deployment of the program are done, difficulties recorded within the spreadsheets.

2.1.2 Configuration Items

Configuration items identified for the project are:

1- The Bride Software Management Plan
2- The Bride Configuration Management Plan
3- The Bride Code Standards
4- The Bride Requirements Data
5- The Bride Design Data
6- Project Environment List
7- Source Code of the Project
8- Executable Object Code
9- Weekly Management Records
10- The Bride Test Specifications
11- The Bride Test Data

A Configuration Item Index shall be created and placed under CVS server. It will be updated whenever a new CI or new version is established and e-mailed to all development team.
Environmental List Form will be used to catalog any electronic media, such as CDs, floopy disks, application software & firmware. This list will be prepared by the team leader and forwarded to SCM for configuration control.

CIs identified in this SCMP shall be under the control of software development and verification team until they are released or baselined, then they will be under configuration control at a controlled library folder using CVS tool by SCM.

"CVS" tool will be used for the configuration control and maintaining the changes of the project configuration items. Changes are automatically assigned to a revision number. The revision number along with the change is then stored in an associated archive.

2.1.3 Work products and documentation

- Identify change: Once the change is identified a Change Request Form will be produced and will be sent to all the members of the SCM team.

- Control change: After evaluator (SCM team member) got the change request form, Change Report Form will be generated.
- Ensure that change is being properly implemented.
- Document the change: Once the change is approved we will document the change in the library. And we will change the software version number if it is necessary.

2.2 Configuration Control

Changes will be controlled automated tools & contact with the team members. Here are the steps, which will be taken in order to control change.

- Request the change
- Software developer will evaluate the change request
- The result of the evaluation will be presented as change report
- Final decision on change will be made
- If change is approved

1. Define constraint
2. "Check out" items for changes
3. Make necessary change
4. Apply SQA activities
5. "Check in" items
6. Apply testing activities
7. Rebuilt the software
8. Distribute the software

2.3 Version Control

2.3.1 Description

As a result of changes, the version number of various modules will be increased accordingly. We will be using a universal version number system for all modules. We will also have a final version of the entire product.

Major documentation will also have version numbers, such as User Manual or Design Specification.

2.3.2 Version Number

When a change request is filed, a change report will be created. After the change is finalized, it will be documented in the library. We will be using CVS tool for controlling the version of the program.

Bug Fixes

If enough bug fixes have been done on the product/module, the bug fix portion of the version number will be increased. The number of user visible bug fixes will also affect when the bug fix number is increased. The more visible bug fixes have been made, the closer the bug fix number will need to be increased.

Minor Update

If functionality is added to the product/module that will increase the user-friendliness / performance but does not change the way a function/interface work, the minor update number may be increased. Several of these changes will warrant a version number change. Again, visible changes (interface) will cause the version number to increase sooner.
If a major functionality has been added to the product that greatly increase the user's experience or greatly improves the program performance, a minor version update will be issued immediately.

Major Update

When there is a major change in the product the new version of the program will be the increased value of the older version.

2.3.3 Work Products and Documentation

All the products related to the version control will be accessible through the use of version control system CVS.

3. Environment

The following subsections describe the environment in which the tools to be used satisfy the objectives of the SCMP.

3.1 CVS

CVS shall be used to maintain changes made to Configuration Items (CIs) throughout the software development life cycle. Changes are automatically assigned a revision number. The revision number along with the change is then stored in an associated archive. Any revision of a file may be retrieved at any time. Specific revisions shall be version tagged in order to retrieve specific builds and baselines.

This tool shall be used for the following activities:

- Configuration Identification
- Configuration Status Accounting
- Software Life Cycle Data Controls
- Version Control and Retrieval

3.2 Microsoft Excel

Excel spreadsheets will be used to catalog any electronic media, such as CDs, floppy disks, application software, etc and any hard copies of documentation for which we have no soft copies. This spreadsheet records the uniquely assigned identification number, title and description of the item, date of the item, version of the item (if applicable), date received into CM, and it's physical or soft copy location.

This tool shall be used for the following activities:

- Configuration Identification

- Configuration Status Accounting

- Archive, Retrieval and Release

- Software Life Cycle Environment Control

4. Configuration Status Accounting (CSA)

The objective of this activity is to provide data for configuration management of software life cycle processes with respect to configuration identification, baselines, SCRs, and change control.
The output of this facility is Software Configuration Management Records, The Bride.

4.1 Description

The team members will be communicating with each other or the people associated with software development & get the related data by these ways:

- Change request report: We will have two different forms that we will use as tools to request a change or to report a change to the software configuration manager. These documents will be in html format and we will be able to send them through web.
- Verbal communication: Since our software development team is small and all the team members are in touch with each other it would be better to communicate verbally.
- CVS to hold all the related data about changes, old versions, baselines.

4.2 Work products and documentation

- Change request reports
- Emails
- Test errors will be documented through the use of excel spreadsheets.
- All suggestions made during peer reviews, and internal project reviews will be noted

5. Software Quality Assurance Overview

SQA activities will focus on the management issues and the process specific activities that enable a software organization to ensure that it does “the right things at the right time in the right way.” The Bride SQA plan provides a road map for instituting software quality assurance.

5.1 Scope and Intent of SQA Activities

The objectives of SQA are:

- A quality management approach

- Effective software engineering technology (methods and tools)
- Formal technical reviews that are applied throughout the software process
- A multi testing strategy is drawn
- Control of software documentation and the changes made
- A procedure to assure compliance with software development standards when applicable
- Measurement and reporting mechanisms

5.2 Reviews and Audits

A formal technical review (FTR) is a software quality assurance activity that is performed by software engineers. The objectives of the FTR are:

1- Uncovering errors in function, logic, or implementation for any representation of the software;
2- Verifying that the software under review meets its requirements;
3- Ensuring that the software has been represented according to predefined standards;
4- Achieving software that is developed in a uniform manner;
5- Making projects more manageable.

5.2.1 Generic Review Guidelines

The following are the guidelines that we are going to follow through SQA facilities.

- Management Responsibility
- Design Control
- Document and Data Control
- Product Identification and Trace Ability
- Process Control
- Inspection and Testing
- Control of Inspection, Measuring, and Test Equipment
- Inspection and Test Status
- Control of Nonconforming Product
- Corrective and Preventive Action
- Handling, Storage, Packaging, Preservation, and Delivery

Conduction of a Review

All reviews will be done with the attendance of all the team members. For the changes that will affect the performance of the software the entire team members has to agree with the change and keep a good record of the project before and after changes.

Roles and Responsibilities

As stated in SQA Organizational Role, the rule of each team member will be very confusing since we have a relatively small team. All team members will take their place effectively in the development phase.

Review Work Product

For each period (weekly, in our case), we'll generate a work report from each member. In the work report, we will state each member's work for the past week, problems that encounter, problem that can't be solved, any cautions to remind. This report will be extremely helpful when comes to documentation and writing the help menu.

5.2.2 Formal Technical Reviews

Here are the FTR that we will conduct during the software process:

- Walkthroughs
- Inspections

After each baseline for the source code, we'll do a test on the component using black box testing method. And for each week, when the team set down come to a meeting, we will ask the team mates to do an inspection on the interface, then hook up the other's work, do a walkthroughs of all the components.

Description of review Walkthroughs

Reviews mainly focus on the integrations of the parts that we develop. We will ask other team members to do the walkthroughs with the presence of coder.

Description of Review Inspection

This review is mainly focus on the correctness of the parts that we designed. Usually allow the other team-members to do a private test, without the designer's interrupt. This idea is try to allow other team-members bring out the test cases.

System Specification Review

System specification usually changed after each weekly meeting, and after each meeting. As for this moment, most of the system design have been settle down. Even if we have expended the project into a higher level, the basic mapping of the project is still the same.

Software Project Plan Review

The purpose of Software Project Plan is over look of the whole project. For more information, please see the document titled "Software Project Plan".

RMMM Review

RMMM, Risk Mitigation, Monitoring & Management, is use to prevent, monitor, and manage the risk. For more information, please see the document titled "RMMM".

Requirements Reviews (Models, Specification)

Software Requirements stated the data requirement, specifications. For more information, please see the document as "Requirement Analysis Report".

Data Design Review

The Data Design document is about the data flow between each module. For Data Design, please see the document "The Bride Final Design".

Architectural Design Review
The Architectural Design document is about the whole project design, layout, and data flow. For Architectural Design, please see the document titled “The Bride Final Design”.

Component Design Review

Interface (GUI)
The entry point for the game shall be updated when there is a difficulty due to the internal components of the program.

Code Review

Test Specification Review

Change Control Reviews and Audits
All changes shall be reviewed by different team members than the one makes the updates, and approved by the team leader before the new life cycle data is released, in the formal phase.

5.3 SQA Audits

Team members will have a weekly report on their individual performance for the past week. Any problems, question regardless on the performance of other team members will also be noted there.
Any changes that will affect the project will consult with other team members before doing any changes. These are the changes that are minor or require little code change, but still different from the original architectural design.

6. APPENDIX

6.1 Configuration Data

Software life cycle data related to the configuration management controls are as follows:

1. Project Plans
2. Software Coding Standards
3. Software Requirements Data
4. Software Design Data
5. Software Configuration Identification Index
6. Software Life Cycle Configuration Environmental Index (Environmental List)
7. Source Code
8. Executable Object Code
9. Software Test Cases, Procedures, and Results
10. Accomplishment Summary
11. Software Configuration Management Records

6.2 Related Documents

- The Bride Software Management Plan
- The Bride Software Configuration Management Plan.
- The Bride Requirements Analysis Report
- The Bride Final Design Report
- The Bride Software Code Standards
- The Bride Test Specifications
- Version Control with CVS Instruction